Theoretical Computer Science 48 (1986) 75–94
North-Holland

A UNIFIED THEORY OF INTERCONNECTION NETWORK STRUCTURE*

Clyde P. KRUSKAL
Department of Computer Science, University of Maryland, College Park, MD 20742, U.S.A.

Marc SNIR
Institute of Mathematics and Computer Science, Hebrew University, Jerusalem 91904, Israel

Communicated by E. Shamir
Received October 1983
Revised March 1984, June 1986

Abstract. The relationship between the topology of interconnection networks and their functional properties is examined. Graph-theoretical characterizations are derived for delta networks, which have a simple routing scheme, and for bidelta networks, which have the delta property in both directions. Delta networks are shown to have a recursive structure. Bidelta networks are shown to have a unique topology. The definition of bidelta network is used to derive in a uniform manner the labelling schemes that define the omega networks, indirect binary cube networks, flip networks, baseline networks, modified data manipulators and two new networks; these schemes are generalized to arbitrary radices.

The labelling schemes are used to characterize networks with simple routing. In another paper (Kruskal/Snir, 1984), we characterize the networks with optimal performance/cost ratio. Only the multistage shuffle-exchange networks have both optimal performance/cost ratio and simple routing. This helps explain why few fundamentally different geometries have been proposed.

Key words. Banyan network, baseline network, bidelta network, capacity, delay, delta network, flip network, indirect binary cube network, interconnection network, isomorphism, multistage network, omega network, packet-switching network, routing, topological equivalence.

1. Introduction

There has been large amount of research on multistage interconnection networks for parallel processing (see, for example, [4, 6, 15, 17]). Nevertheless, there seems to be a surprisingly small number of basic designs for interconnection networks

* This work was supported in part by the Applied Mathematical Sciences Program of the U.S. Department of Energy under contract No. DE-AC02-76ER03077, in part by the National Science Foundation under Grant Nos. NSF-MCS79-21258 and NSF-MCS81-05896, and in part by a grant from IBM. The work was done while the first author was at the University of Illinois, Urbana, IL 61801, U.S.A. and the second author was at the Courant Institute, New York University, New York, NY 10012, U.S.A. Some of this material appeared in the *Proc. 1982 Conf. on Information Sciences and Systems*, Princeton University, March 1982.

that recur under many disguises. A particularly ubiquitous geometry is the 'multistage shuffle-exchange network' [1, 2, 5, 7, 11, 12, 14, 16, 19]. This geometry provides good performance and 'simple' message routing (or control). Given the paucity of other network geometries, one is tempted to conjecture that these networks are in some sense optimal. In this paper, we shall characterize the networks with simple message routing. In another paper [10], we characterize the geometries that provide optimal performance/cost ratio. The multistage shuffle-exchange networks are the unique networks with both simple routing and optimal performance/cost ratio. This helps explain why few fundamentally different geometries have been proposed.

The approach followed in this paper differs from that followed in most of the literature on interconnection networks. Most research concentrates on the properties of specific networks. In this paper we shall start from a set of desired properties and obtain a complete description of the networks having those properties. Starting from a functional definition for 'delta' networks [13], i.e., networks in which routing is done according to the successive bits of the destination, we obtain a complete description of the possible geometries for such networks. The geometry of 'bidelta' networks, i.e., networks in which the delta property holds in both directions, is shown to be unique. The labelling schemes that define omega networks, indirect binary cube networks, flip networks, baseline networks, reverse baseline networks, modified data manipulators, and two new networks are derived in a uniform manner.

2. Definitions

An *(M, N)-network* G is a directed graph whose nodes include a set I of M distinguished *input* nodes and a set O of N distinguished output nodes. We shall use *switch* as a synonym for node. We assume, without loss of generality, that for each node u of G there is a directed path connecting some input to u and a directed path connecting u to some output. The *indegree* of a node is the number of edges leading into it, and the *outdegree* of a node is the number of edges leading out of it.

A network with a unique path from each input to each output is called a *banyan network* [7]. A banyan network is *layered* if the nodes can be arranged in successive layers, with inputs at the first layer, outputs at the last layer, and edges connecting nodes from one layer to nodes at the next layer. A layered network with n stages of nodes is called an *n-stage network*. A *rectangular banyan network of degree k* is a layered banyan network where all nodes (with the exception of inputs) have indegree exactly k, and all nodes (with the exception of outputs) have outdegree exactly k. A rectangular banyan network has $n+1$ layers, each consisting of k^n nodes.

Components of a network must be labelled in order for routing of messages to be possible: the outputs of the network need to be labelled by distinct *addresses*; the edges going out of each node have to be distinguished by a numbering scheme; and if the network is centrally controlled, each node has to carry a distinct label.

We shall assume that if a node has k outgoing edges, then these edges are numbered from 0 to $k-1$.

Two networks are *topologically equivalent* if their underlying graphs are isomorphic; they are *isomorphic* if there exists a label-preserving graph isomorphism between them. The two networks in Fig. 1 are topologically equivalent since, ignoring labels, they look exactly alike; but the networks are not isomorphic.

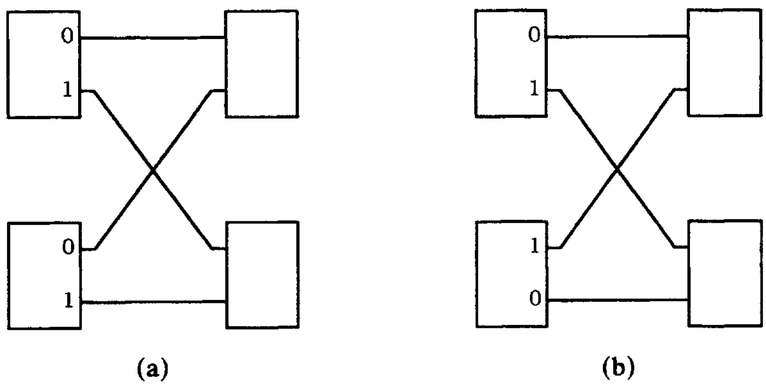

Fig. 1. Two networks that are topologically equivalent but not isomorphic.

A path from an input to an output can be described by the sequence of labels that label the successive edges on this path. We shall call this string the *path descriptor.* In a message-switching network with nonadaptive routing, the path descriptor may be used as a header for routing a message: each successive node uses the first element of the string to route the message, and then discards it.

3. Delta networks

What labelling scheme will simplify routing? In a general network, the paths leading from different input nodes to the same output node may have different path descriptors. Thus, a routing table, containing a path descriptor for each output node, is needed at each input node. It is convenient to have all these tables identical. Then we can take the path descriptor associated with paths leading to the output node s to be the *address* of s, and the unique information needed to route a packet to an output node is the address of that node.

Extending the original definition of Patel [13], we define a *digit controlled* or *delta network* to be a network with the following properties:

(1) there is a path from each input node to each output node;

(2) the path descriptors associated with paths leading to the same output node are identical.

The second condition implies that there is at most one path from an input to an output since two different paths from the same input have different descriptors. This, along with the first condition, implies that delta networks are banyan networks, i.e., there is exactly one path from each input to each output. The second condition also implies that all the paths leading from the inputs to any particular node in the network have the same length. In particular, the nodes of a delta network can be arranged by stages so that the input nodes are all at stage 1, and edges connect nodes at stage i only to nodes at stage $i+1$. Note, however, that the output nodes of a delta network need not all be at the same stage—see Fig. 2. Delta networks in which all outputs are at the same stage are layered banyan networks (according to the definition given in Section 2).

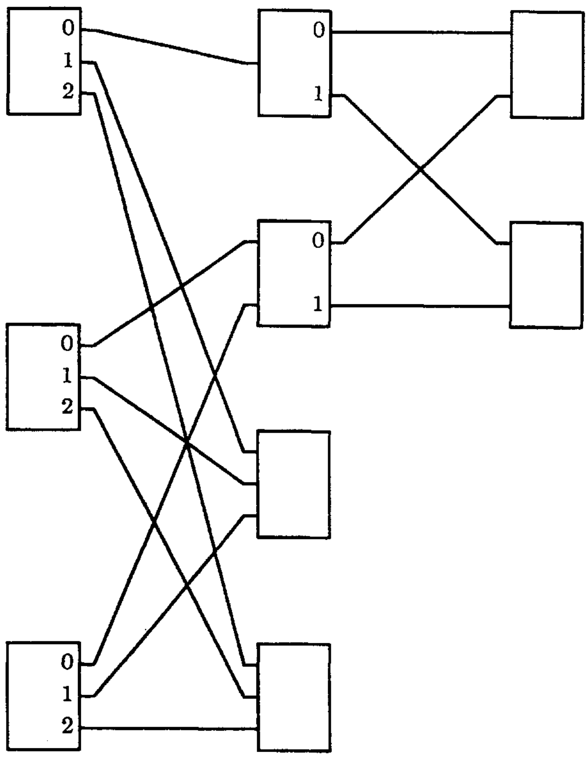

Fig. 2. Delta network.

Let $G=\langle V, E\rangle$ be a network with inputs I and outputs O, and let S be a subset of nodes from G. The *subnetwork* $G'=\langle V', E'\rangle$ *spanned by* S is the network whose graph consists of all nodes and edges reachable by some (directed) path starting from any node in S (including the nodes in S). A node $i\in V'$ is an input node of G' if no edge of E' enters i; a node $o\in V'$ is an output node of G' if no edge of E' exits o. (Input nodes of G' will be nodes from S; output nodes of G' will be nodes from O.)

If G is a delta network, then a delta network structure is induced on 'reasonable' subnetworks of G. The concept of a 'reasonable' subnetwork is embodied by the conditions of the following lemma.

Lemma 3.1. *Let G be a delta network, and let G' be the subnetwork spanned by a set S of nodes in G. Then G' is a delta network if it fulfills the following two conditions:*

(1) *there exists a path from each input to each output in G';*

(2) *all the inputs of G' are nodes from the same stage of G.*

Proof. Suppose G' fulfills both conditions. Condition (1) is the first condition definining delta networks, so G' is a delta network if the path descriptors associated with paths leading to the same output are identical. Let p and p' be descriptors for paths from two distinct inputs s and s' of G' to the same output t of G'. Let q and q' be descriptors of paths leading from inputs of G to s and s' in G, and let $|q|$ and $|q'|$ be their lengths. Then $qp = q'p'$, and $|q| = |q'|$, so that $p = p'$. □

The last lemma implies an alternative, recursive definition of delta networks.[1]

Theorem 3.2. *A delta network G either consists of a unique node, or consists of one stage of nodes all with same outdegree k, followed by k (disjoint) delta networks $G_0, \ldots, G_{k-1}$; each node in the first stage is connected to an input of G_i via an edge with label i, for $i = 0, \ldots, k-1$.*

Proof. Let u be a node of the delta network G. All paths connecting inputs of G to u have the same path descriptor: indeed, let i and i' be two inputs connected to u by paths with descriptors α and α', respectively; let u be connected to an output o via a path with descriptor β. Then $\alpha\beta = \alpha'\beta$ so that $\alpha = \alpha'$.

Let G_i be the subgraph induced by the set of nodes in the network that are reached by paths starting with label i (Fig. 3). The previous remark implies that these subgraphs are disjoint; by Lemma 3.1, they are delta networks. □

Conversely, it is easy to see that any network that can be recursively decomposed as specified in Theorem 3.2 is a delta network. We have thus achieved a geometric (recursive) definition of delta networks.

Not every rectangular banyan network admits a decomposition of the form given above and, therefore, not every rectangular banyan network can be labelled to be a delta network (for example, see Fig. 4). On the other hand, it immediately follows

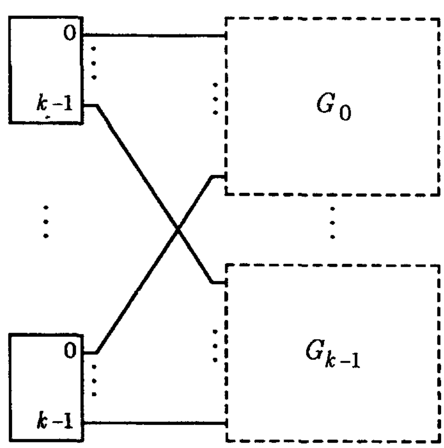

Fig. 3. Recursive structure of delta network.

[1] This theorem was independently proved by Dias [3], and by Kruskal and Snir [9].

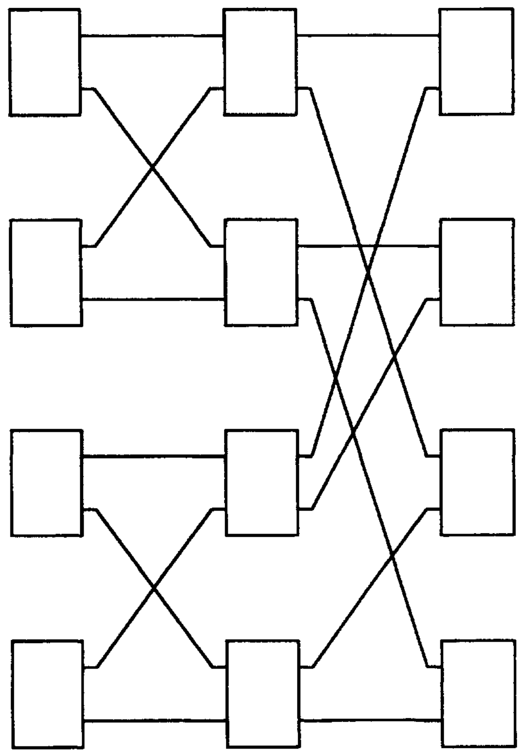

Fig. 4. Nondelta rectangular banyan network.

from the recursive definition that for each k and n there exists an n-stage rectangular delta network of degree k. This rectangular delta network is not unique: Figure 5 shows two 3-stage rectangular delta networks of degree 2 that are not even topologically equivalent.

4. Bidelta networks

In many applications, traffic through the network is bidirectional. It is convenient then that the traffic in the reverse direction also traverses a delta network—especially if the 'output' nodes can initiate requests. The *reversal* G^R of the network G is the network obtained from G by reversing the direction of each edge, and replacing each input by an output and each output by an input. We assume that the reverse network G^R is labelled or, equivalently, that the inputs of G carry distinct addresses, and that each edge of G carries two numbers, one associated with each of the two incident nodes.

A network G is a *bidelta network* if both G and G^R are delta networks. In a bidelta network all paths connecting inputs to outputs have the same length, all paths leading from inputs to a given node have the same length and the same path descriptor, and all reverse paths leading from outputs to the same node have the same length and the same path descriptor—see Fig. 6. We use the term *multistage*

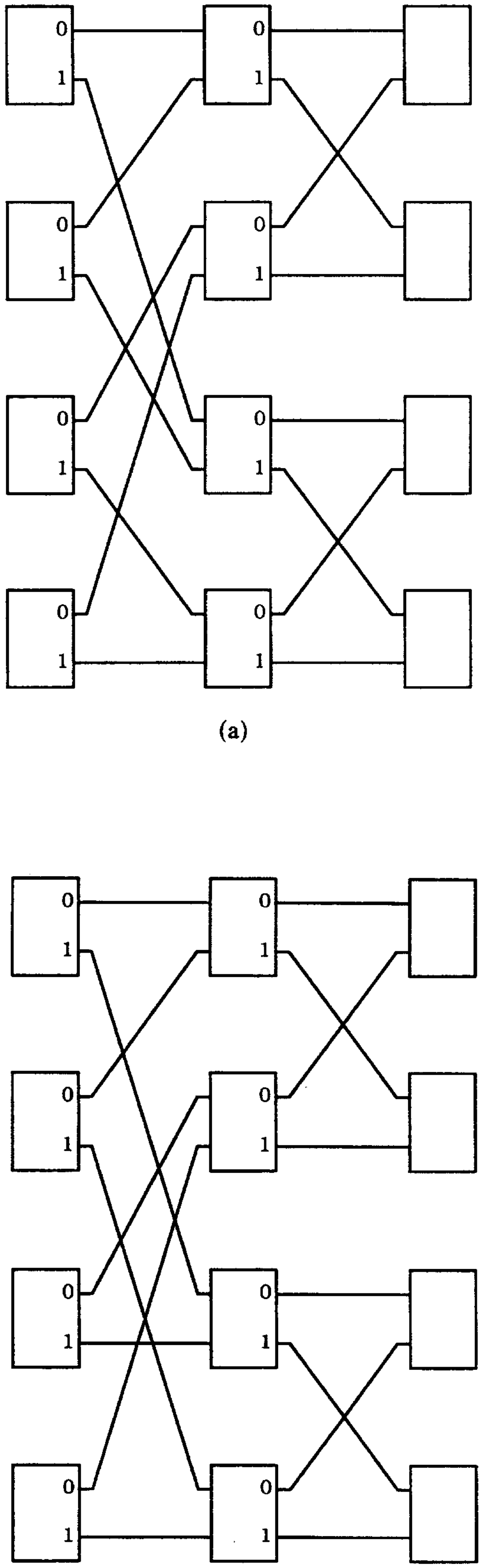

Fig. 5. Two 3-stage delta networks of degree 2 that are not topologically equivalent.

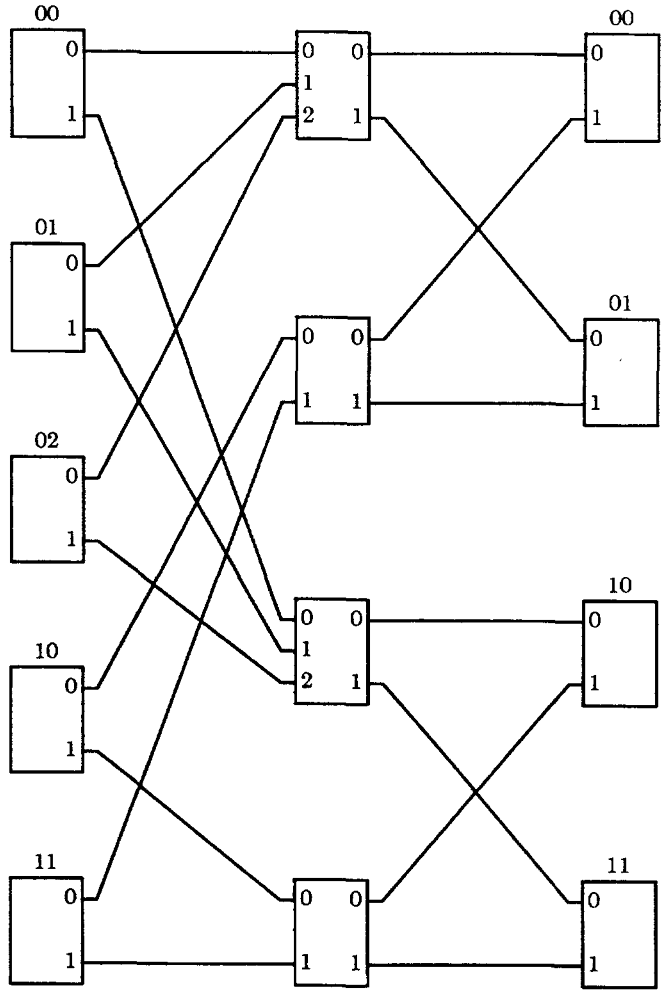

Fig. 6. Bidelta network with addresses.

shuffle-exchange network for any n-stage bidelta network of degree k (although in the literature the term often refers only to networks of degree 2).

In packet-switching networks, when messages are sent from inputs to outputs, replies are often returned to the sender. We remark, in passing, that in a packet-switching network that has labels on both the input edges and the output edges, a message need not (initially) carry the sender address [8, 18]. Rather, this address can be created on the fly when the message is routed: whenever one digit from the 'forward' path descriptor is discarded, it is replaced by one digit that identifies the edge through which the message has arrived. When the message arrives at its destination, it carries a path descriptor for the reverse path to the sender; in a bidelta network this is the address of the sender.

Two bidelta networks are isomorphic if there is a label-preserving graph isomorphism between them (labels for both directions are preserved). Now we achieve at last our hope of having a functional description that defines a unique network[2]

[2] This theorem was independently proved by Dias [3], and by Kruskal and Snir [9].

Theorem 4.1. *Any two n-stage bidelta networks of degree k are isomorphic.*

Proof. The claim is trivial for $n=1$. Assume it holds for $n-1$, and let G and G' be two n-stage bidelta networks of degree k. As G^{R} and G'^{R} are bidelta networks, they both admit a decomposition of the form illustrated in Fig. 7, where G_i^{R} ($G_i'^{\mathrm{R}}$), $i=0,\ldots,k-1$, are $(n-1)$-stage delta networks of degree k, and each output of G_i (G_i') is connected to an output of G (G') in the last stage via an edge labelled at its head with i. Since G is a delta network, the k edges connected to the same output of G (G') have the same label at their tails. By Lemma 3.1, each network G_i, G_i' is a delta network. It follows that each of them is an $(n-1)$-stage bidelta network, and therefore, all of them are isomorphic. The claim now follows. □

The theorem can be extended to nonrectangular bidelta networks: for any tuple $\langle 0=p_1, q_1, p_2, q_2, \ldots, p_{n-1}, q_{n-1}, p_n, q_n=0\rangle$ there exists, up to isomorphism, a unique bidelta network consisting of n stages of nodes, where nodes at stage i have indegree p_i and outdegree q_i. Thus, the functional properties of bidelta networks

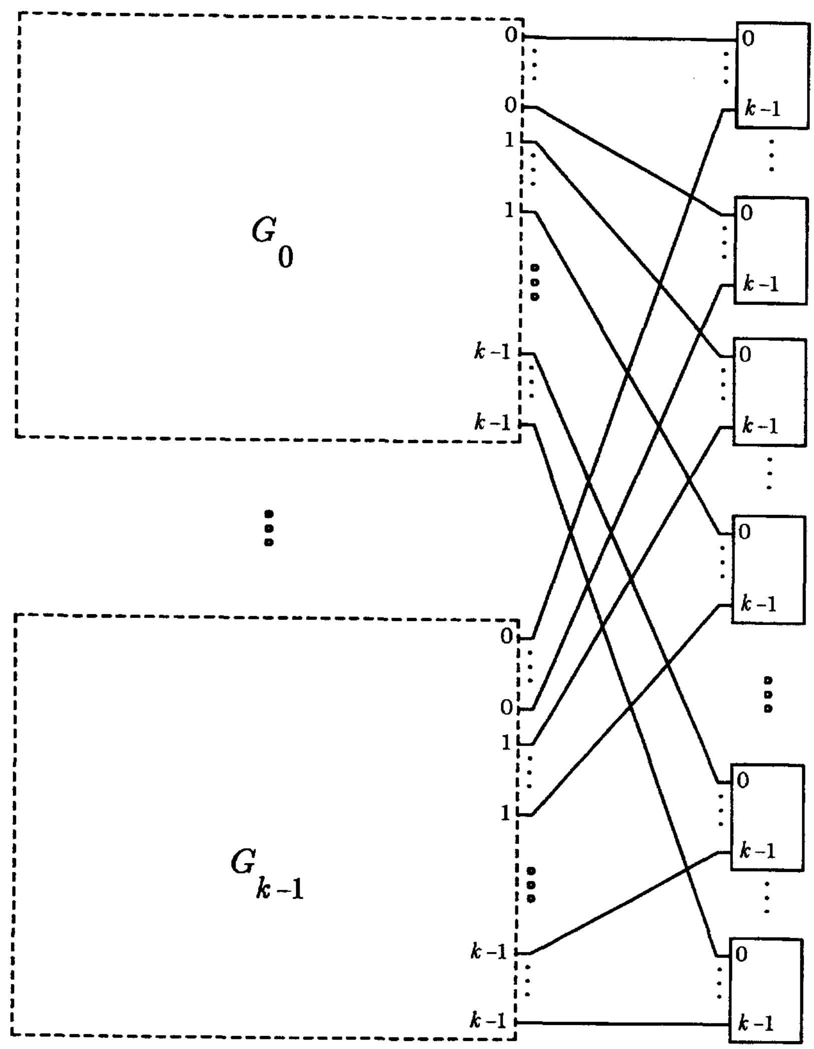

Fig. 7. Decomposition of reverse bidelta network.

uniquely determine the structure of such networks once the size of the network and the indegree and outdegree of the nodes at each stage are fixed.

5. Node numbering scheme

In bidelta networks, the path descriptors on both directions can be used to uniquely identify any node in the network. For each node s of G, denote by $\vec{\alpha}(s)$ the path descriptor of the paths connecting inputs to s, and denote by $\overleftarrow{\alpha}(s)$ the path descriptor of the reverse paths connecting outputs to s. (We omit s when it can be inferred from the context.)

Lemma 5.1. *Let G be a bidelta network. Node $s = s'$ iff $\vec{\alpha}(s) = \vec{\alpha}(s')$ and $\overleftarrow{\alpha}(s) = \overleftarrow{\alpha}(s')$.*

Proof. Suppose $\vec{\alpha}(s) = \vec{\alpha}(s')$ and $\overleftarrow{\alpha}(s) = \overleftarrow{\alpha}(s')$. Let u and u' be inputs connected to s and s', respectively, and let v and v' be outputs connected to s and s', respectively. The path connecting u' to v has the same path descriptor as the path connecting u to v. This path descriptor starts with $\vec{\alpha}(s)$, which is also a descriptor for the path from u' to s'. The path from u' to v passes, therefore, through s'. A similar argument shows that the reverse path connecting v' to u passes through s'. Node u is connected to v by a path that passes through s', as well as by a path that passes through s (see Fig. 8). Thus, $s = s'$.

The converse implication is immediate. $\square$

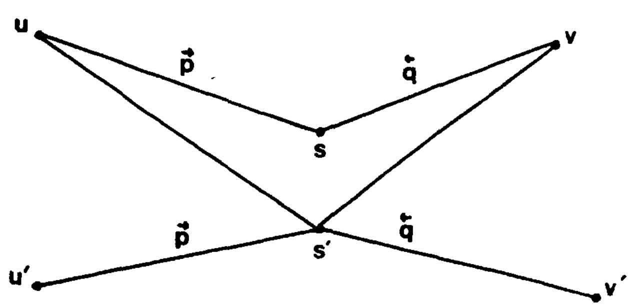

Fig. 8.

The last result implies that the following naming convention can be used to uniquely identify each node in a bidelta network G: associate with each node s of G an address $\alpha(s) = \vec{\alpha}(s)\overleftarrow{\alpha}(s)$ obtained by concatenating the path descriptor of the paths connecting inputs to s and the path descriptor of the reverse paths connecting outputs to s. Note that this naming convention is consistent with the scheme previously introduced to label output (input) nodes. The naming scheme is precisely that which describes the structure of the *baseline network* [19]. It is illustrated in Fig. 9(a) for nodes with indegree and outdegree two.

There are five other (full labelled) networks that are known to be isomorphic to baseline networks. These are the reverse baseline networks [19], the omega networks [11], the flip networks [1], the modified data manipulators [5], and the indirect binary cube networks [14]. (See [12, 16, 19] for a proof of their isomorphism.)

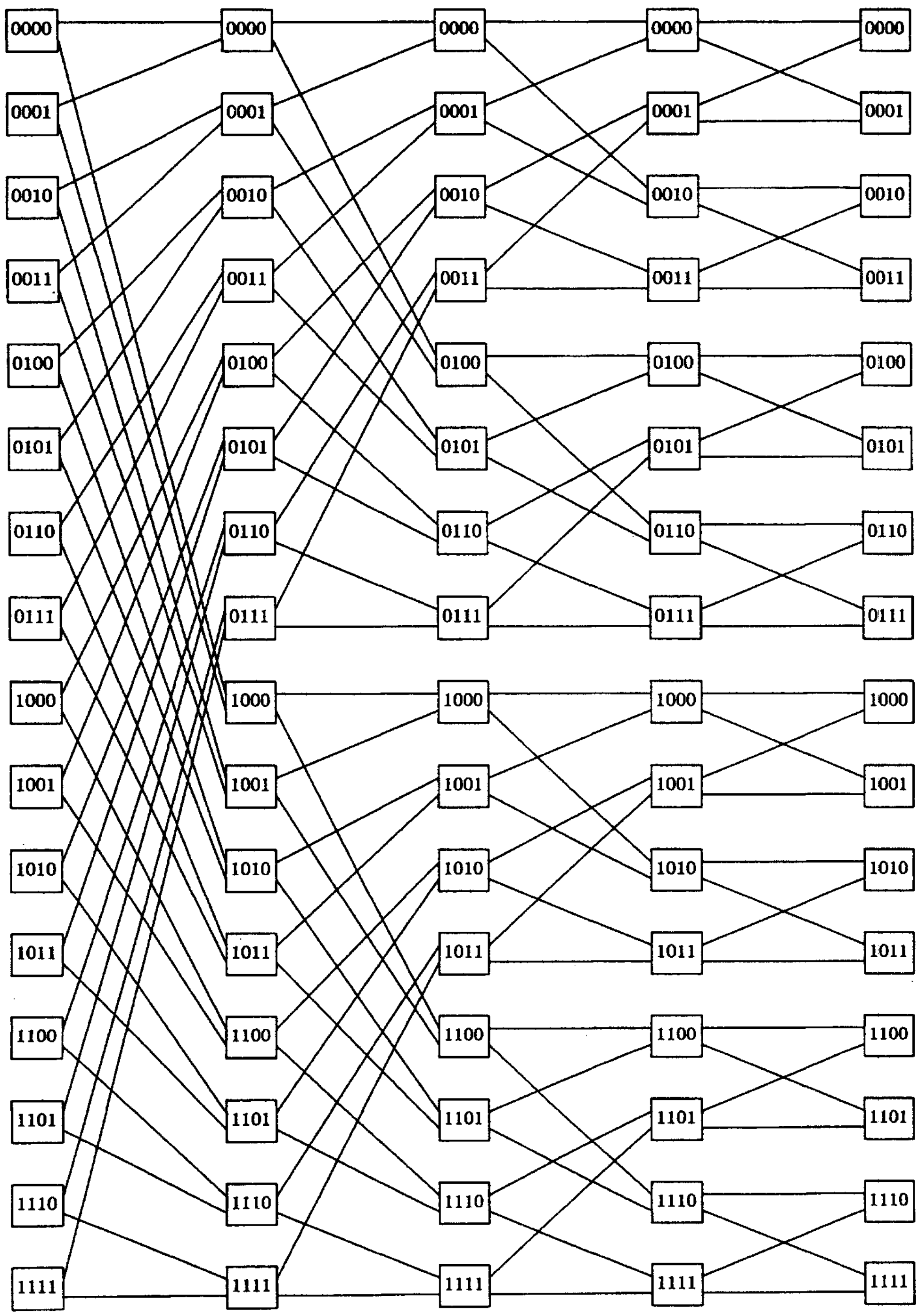

Fig. 9(a). Baseline network; forward path descriptor first.

It turns out that these networks are obtained by slightly varying the naming scheme that yields the baseline network: the two address parts may be exchanged, and each of them may be reversed (Figs. 9(b)-(f)). If a node s is connected to a node s' via an edge that has label δ at its tail and label ε at its head, then $\vec{\alpha}(s') = \vec{\alpha}(s)\delta$ and $\bar{\alpha}(s) = \bar{\alpha}(s')\varepsilon$.

(a) The baseline naming scheme associates to each switch the address $\vec{\alpha}\bar{\alpha}$. Thus, in the baseline network, a node u at stage i with label $\alpha_1, \ldots, \alpha_{n-2}, \varepsilon$ is connected

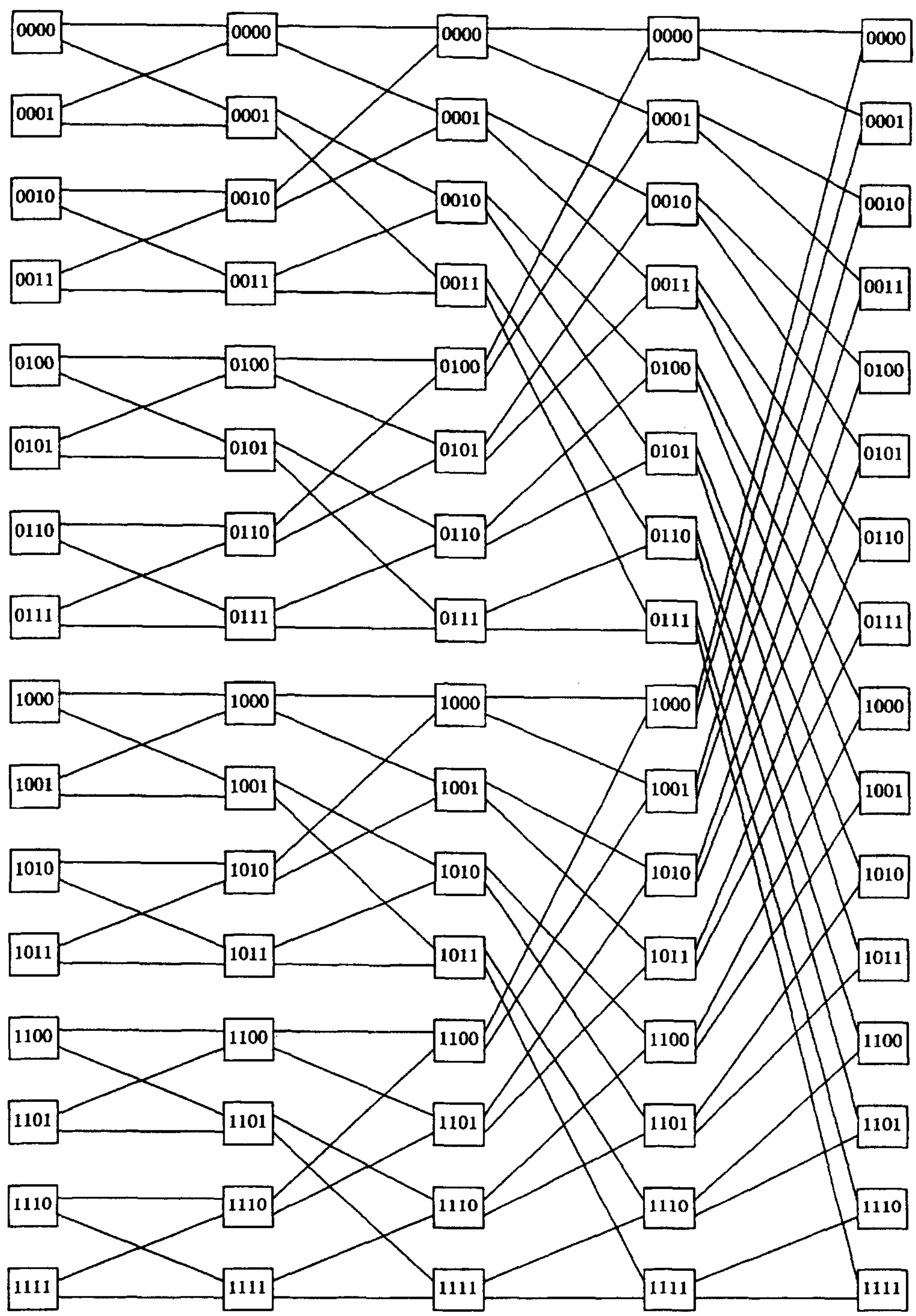

Fig. 9(b). Reverse baseline network; back path descriptor first.

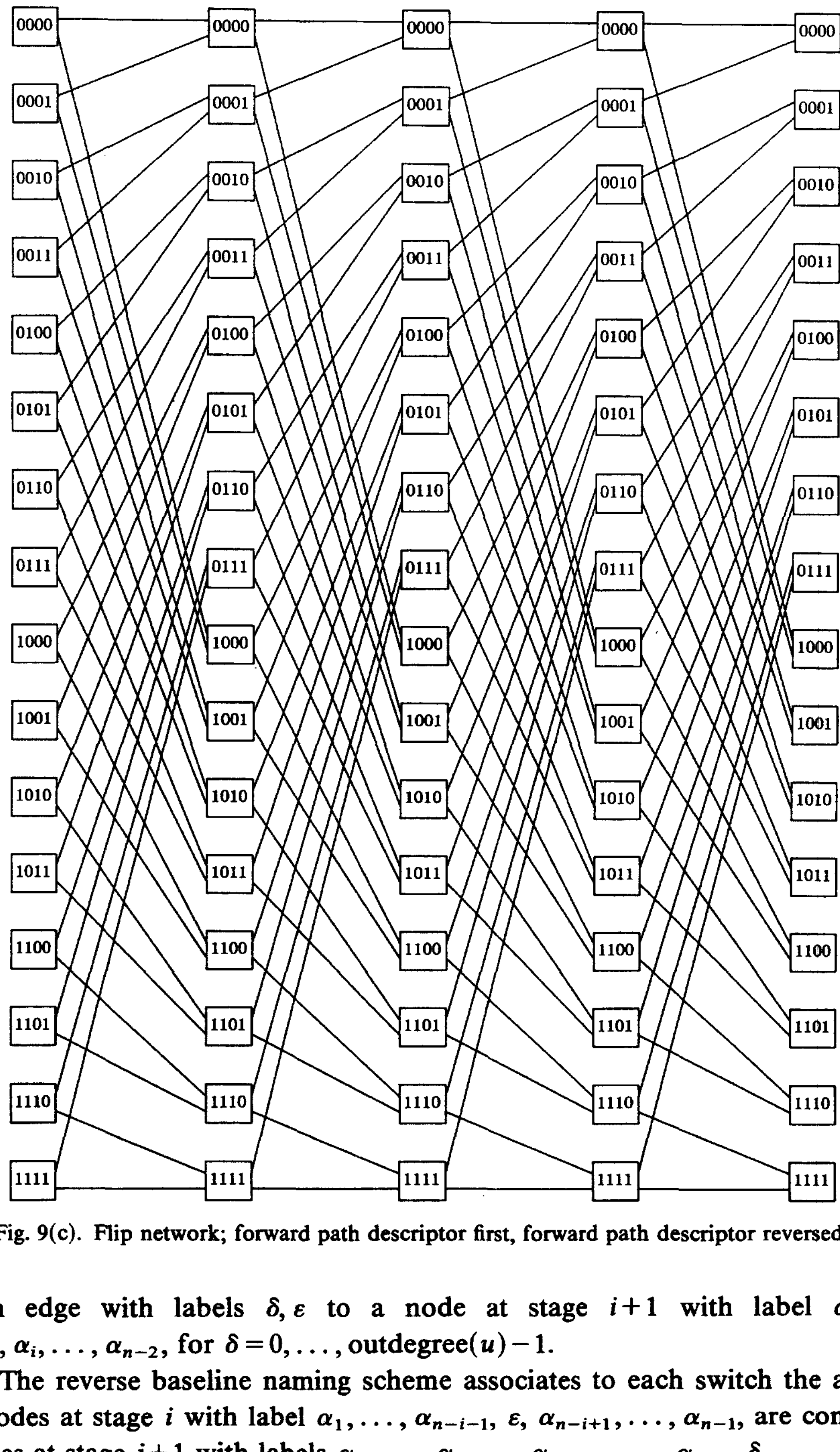

Fig. 9(c). Flip network; forward path descriptor first, forward path descriptor reversed.

via an edge with labels δ, ε to a node at stage $i+1$ with label $\alpha_1, \ldots, \alpha_{i-1}, \delta, \alpha_i, \ldots, \alpha_{n-2}$, for $\delta = 0, \ldots, \text{outdegree}(u) - 1$.

(b) The reverse baseline naming scheme associates to each switch the address $\bar{\alpha}\vec{\alpha}$. Nodes at stage i with label $\alpha_1, \ldots, \alpha_{n-i-1}, \varepsilon, \alpha_{n-i+1}, \ldots, \alpha_{n-1}$, are connected to nodes at stage $i+1$ with labels $\alpha_1, \ldots, \alpha_{n-i-1}, \alpha_{n-i+1}, \ldots, \alpha_{n-1}, \delta$.

(c) The flip network naming scheme associates to each switch the address $\vec{\alpha}^{R}\bar{\alpha}$. Nodes at stage i with labels $\alpha_1, \ldots, \alpha_{n-2}, \varepsilon$ are connected to nodes at stage $i+1$ with labels $\delta, \alpha_1, \ldots, \alpha_{n-2}$.

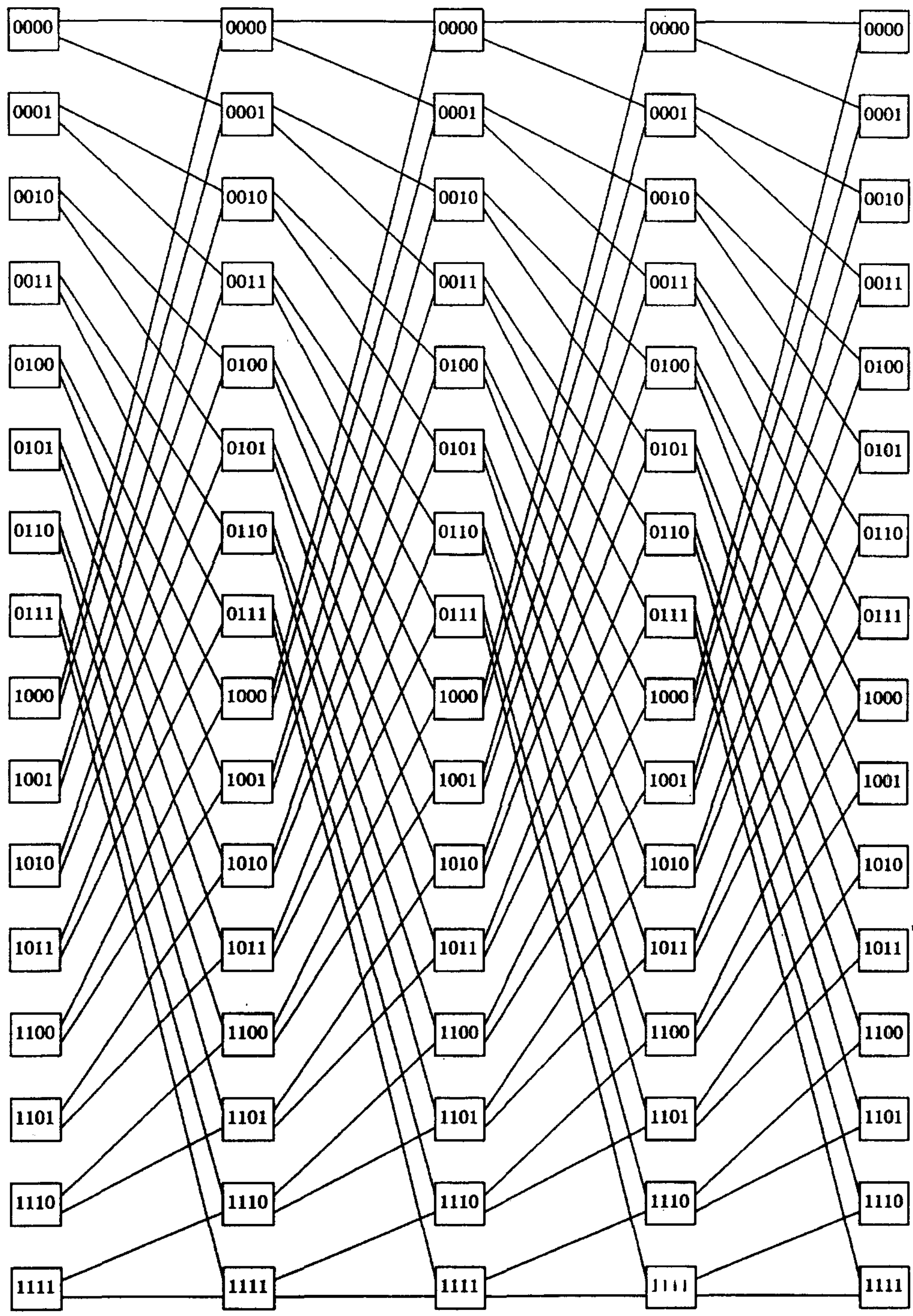

Fig. 9(d). Omega network; back path descriptor first, back path descriptor reversed.

(d) The omega network naming scheme associates to each switch the address $\vec{\alpha}^{R}\vec{\alpha}$. Nodes at stage i with labels to $\varepsilon, \alpha_2, \ldots, \alpha_{n-1}$ are connected with nodes at stage $i+1$ with labels $\alpha_2, \ldots, \alpha_{n-1}, \delta$.

(e) The modified data manipulator naming scheme associates to each switch the address $\vec{\alpha}\vec{\alpha}^{R}$. Nodes at stage i with label $\alpha_1, \ldots, \alpha_{i-1}, \varepsilon, \alpha_{i+1}, \ldots, \alpha_{n-1}$ are connected with nodes at stage $i+1$ with labels $\alpha_1, \ldots, \alpha_{i-1}, \delta, \alpha_{i+1}, \ldots, \alpha_{n-1}$.

(f) The indirect binary cube naming scheme associates to each switch the address $\vec{\alpha}\vec{\alpha}^{R}$. Nodes at stage i with labels $\alpha_1, \ldots, \alpha_{n-i-1}, \varepsilon, \alpha_{n-i+1}, \ldots, \alpha_{n-1}$ are connected to nodes at stage $i+1$ with labels $\alpha_1, \ldots, \alpha_{n-i-1}, \delta, \alpha_{n-i+1}, \ldots, \alpha_{n-1}$.

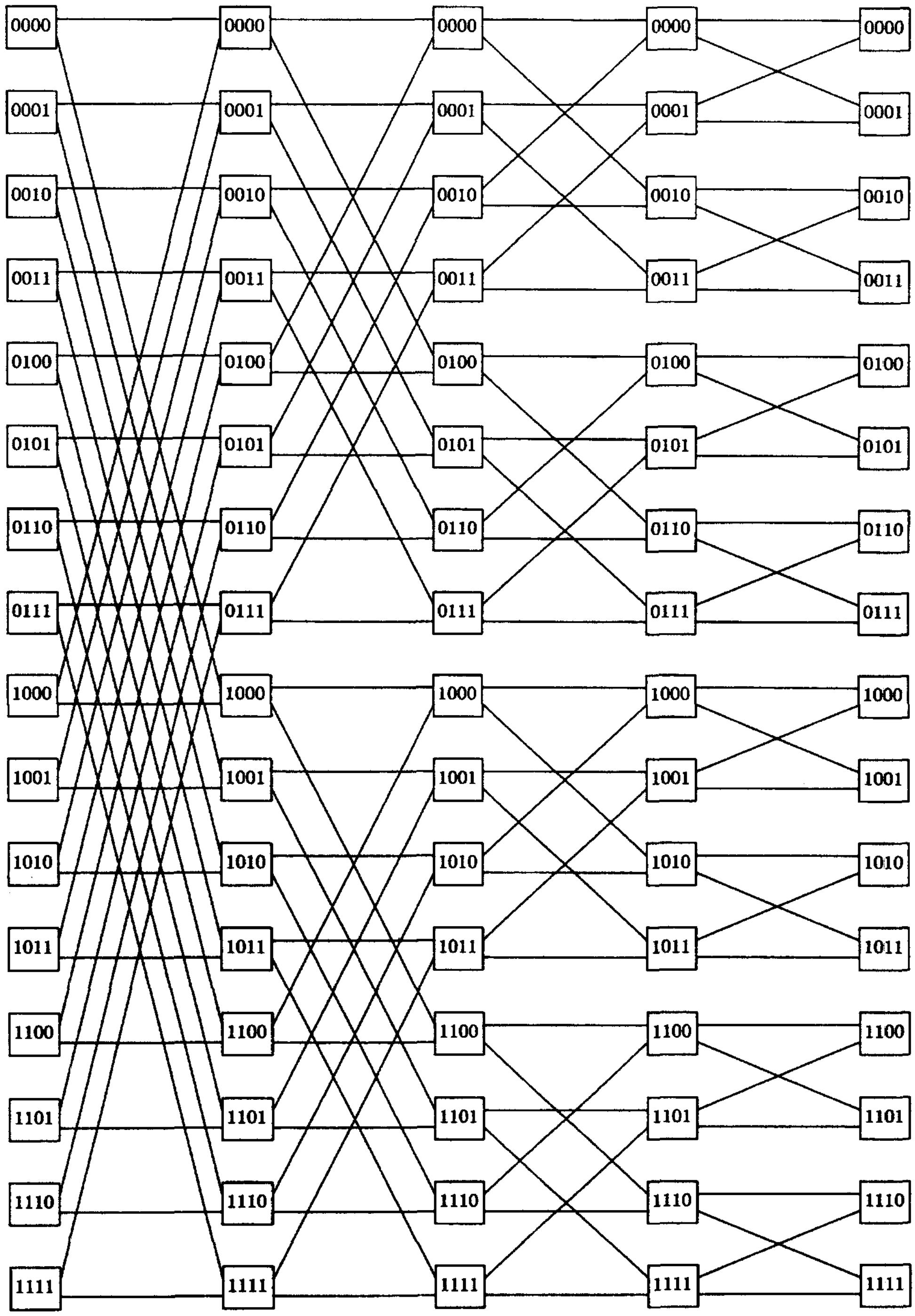

Fig. 9(e). Modified data manipulator network; forward path descriptor first, back path descriptor reversed.

These define the six networks mentioned in [12, 16, 19]. There are two additional networks of the same form, which, alas, remain nameless (Figs. 9(g) and (h)):

(g) If we associate with each switch the address $\vec{\alpha}^{R}\overleftarrow{\alpha}^{R}$, then a node at stage i with label $\alpha_1, \ldots, \alpha_{i-1}, \varepsilon, \alpha_{i+1}, \ldots, \alpha_{n-1}$ is connected at stage $i+1$ to the nodes with labels $\delta, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_{n-1}$.

(h) Finally, if we associate with each switch the address $\overleftarrow{\alpha}^{R}\vec{\alpha}^{R}$, then a node at stage i with address $\varepsilon, \alpha_2, \ldots, \alpha_{n-1}$ is connected at stage $i+1$ to the nodes with labels $\alpha_2, \ldots, \alpha_{n-i}, \delta, \alpha_{n-i+1}, \ldots, \alpha_{n-1}$.

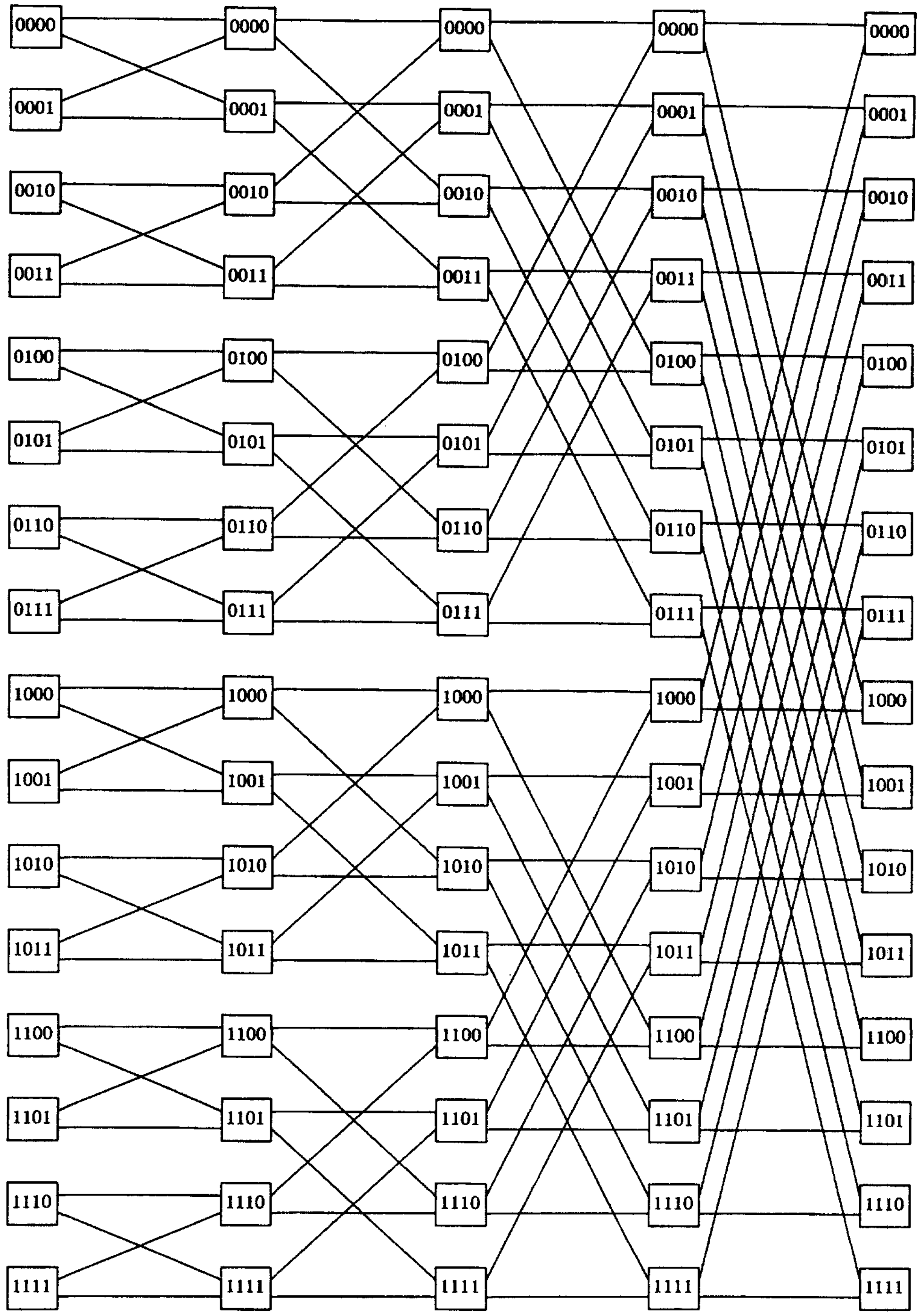

Fig. 9(f). Indirect binary cube network; back path descriptor first, forward path descriptor reversed.

While our Figs. 9(a)-(h) illustrate these eight networks for nodes with indegree and outdegree of two, the definitions are valid for arbitrary indegrees and outdegrees, and even for indegree and outdegree that varies from stage to stage. This immediately extends the definitions of the eight basic networks to arbitrary radix notation.

5.1. Circuit switching

The general node-numbering scheme allows us to uniformly derive the control mechanism for all eight networks when they are used for circuit switching. Given

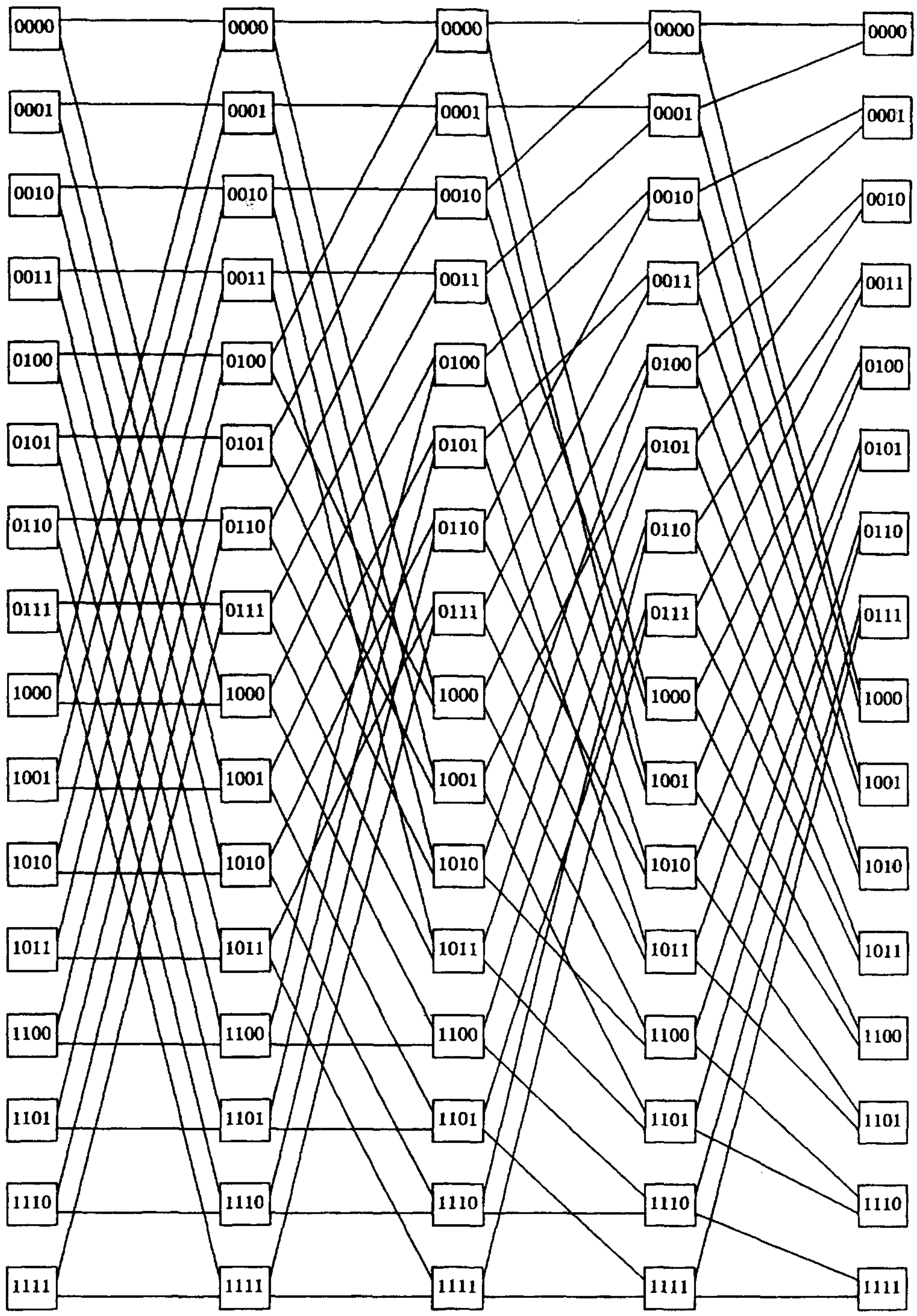

Fig. 9(g). Forward path descriptor first, forward path descriptor reversed, back path descriptor reversed.

a set of input-output connections, one has to determine if the connections do not conflict and, if so, to compute the setting of each switch. If a switch has indegree and outdegree two, then it has two states: 0 (straight) and 1 (cross).

It is convenient to assume that an n-stage network of degree 2 has 2^n input nodes of outdegree 1 and 2^n output nodes of indegree 1, which are not considered switches, and inbetween n stages of switches. This way, all of the switches on a path will be given a setting. Note that the input and output nodes must be appropriately numbered according to the numbering scheme.

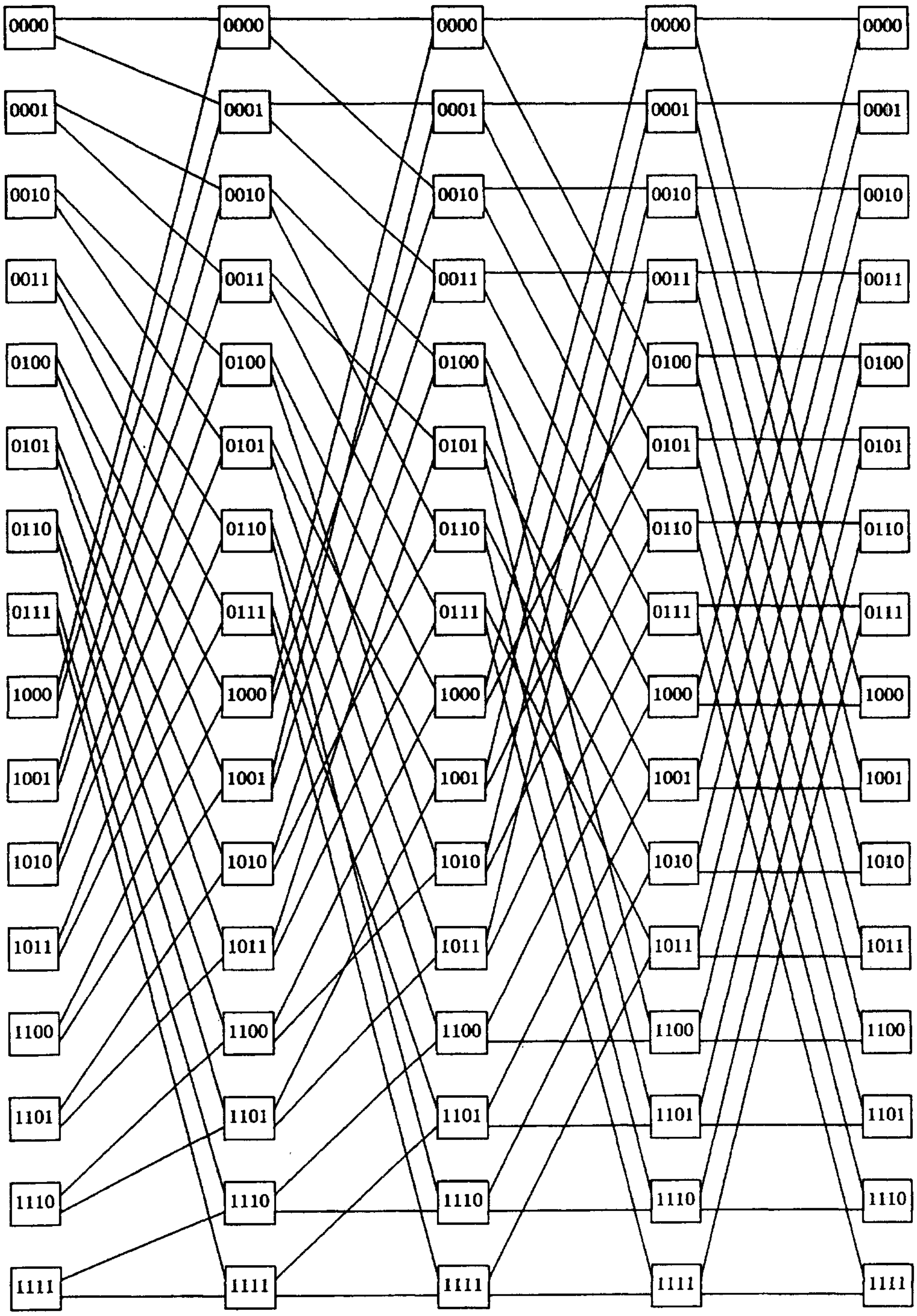

Fig. 9(h). Back path descriptor first, forward path descriptor reversed, back path descriptor reversed.

Suppose that input $\alpha_1, \ldots, \alpha_n$ is connected to output $\beta_1, \ldots, \beta_n$. At stage i, the path crosses the switch with forward path descriptor $\beta_1, \ldots, \beta_{i-1}$ and backward path descriptor $\alpha_1, \ldots, \alpha_{n-i+1}$; it enters the switch via the input labelled with α_{n-i+1} and leaves the switch via the output labelled β_i. If switches of indegree and outdegree 2 are used, then the switch control setting is $\alpha_{n-i+1} \oplus \beta_i$, where $\oplus$ is exclusive-OR.

This can now be used for deriving the control function for each of the previously described eight networks. In each case, a conflict occurs if an attempt is made to set a switch to two distinct states.

(a) Baseline network: at stage i, switch $\beta_1, \ldots, \beta_{i-1}, \alpha_1, \ldots, \alpha_{n-i}$ is set to $\alpha_{n-i+1} \oplus \beta_i$.

(b) Reverse baseline network: at stage i, switch $\alpha_1, \ldots, \alpha_{n-i}, \beta_1, \ldots, \beta_{i-1}$ is set to $\alpha_{n-i+1} \oplus \beta_i$.

(c) Flip network: at stage i, switch $\beta_{n-i+2}, \ldots, \beta_n, \alpha_1, \ldots, \alpha_{n-i}$ is set to $\alpha_{n-i+1} \oplus \beta_{n-i+1}$.

(d) Omega network: at stage i, switch $\alpha_{i+1}, \ldots, \alpha_n, \beta_1, \ldots, \beta_{i-1}$ is set to $\alpha_i \oplus \beta_i$.

(e) Modified data manipulator network: at stage i, switch $\beta_1, \ldots, \beta_{i-1}, \alpha_{i+1}, \ldots, \alpha_n$ is set to $\alpha_i \oplus \beta_i$.

(f) Indirect binary cube network: at stage i, switch $\alpha_1, \ldots, \alpha_{n-i}, \beta_{n-i+2}, \ldots, \beta_n$ is set to $\alpha_{n-i+1} \oplus \beta_{n-i+1}$.

(g) At stage i, switch $\beta_{n-i+2}, \ldots, \beta_n, \alpha_{i+1}, \ldots, \alpha_n$ is set to $\alpha_i \oplus \beta_{n-i+1}$.

(h) At stage i, switch $\alpha_{i+1}, \ldots, \alpha_n, \beta_{n-i+2}, \ldots, \beta_n$ is set to $\alpha_i \oplus \beta_{n-i+1}$.

6. Summary and conclusion

We have presented in this paper a geometrical theory of multistage interconnection networks. We considered labelling schemes for networks. A natural requirement on the labelling scheme, namely that routing be controlled by the successive digits of the address on both directions, was shown to enforce a unique geometry. In this respect, the isomorphism of the omega network, baseline network, flip network, etc., is no mere coincidence, but a result of the functional properties of these networks.

The labelling schemes for the six known bidelta networks and two new ones were derived in a simple, uniform manner. This immediately yielded the algorithms used to control these networks when circuit switching is used.

The following further result shows the close resemblance between delta networks and Benes networks: a network is a Benes network iff it consists of a rectangular delta network G of degree 2 followed by its reversal G^R, where each switch in the last stage of G is identified with the corresponding switch in the first stage of G^R. The proof immediately follows from the recursive characterization of delta networks and from the definition of Benes networks.

This unified theory can be further applied to simplify other results concerning similar interconnection networks, e.g., testing procedures. We hope that this paper will help to prevent the confusion created by the multiplication of notations in this area, and will prevent the duplication of results for these closely related networks.

References

[1] K.E. Batcher, The flip network in STARAN, *Proc. 1976 Internat. Conf. Parallel Processing*, Detroit, MI (1976) 65–71.

[2] L.N. Bhuyan and D.P. Agrawal, Design and performance of a general class of interconnection networks, *Proc. 1982 Internat. Conf. on Parallel Processing*, Columbus, OH, U.S.A.

[3] D.M. Dias, Packet communication in delta and related networks, Ph.D. Thesis, 1981, Rice University, Houston, TX, U.S.A.

[4] D.M. Dias and J.R. Jump, Packet switching interconnection networks for modular systems, *Computer* **14** (1981) 43–54.

[5] T.-Y. Feng, Data manipulating functions in parallel processors and their implementations, *IEEE Trans. Comput.* **C-23** (1974) 309–318.

[6] T.-Y. Feng, A survey of interconnection networks, *Computer* **14** (1981) 12–27.

[7] C.R. Goke and G.J. Lipovski, Banyan networks for partitioning multiprocessor systems, *1st Ann. Symp. on Computer Architecture*, Florida, U.S.A. (1973) 21–28.

[8] A. Gottlieb, R. Grishman, C.P. Kruskal, K.P. McAuliffe, L. Rudolph and M. Snir, The NYU Ultracomputer—designing an MIMD, Shared-Memory Parallel Machine, *IEEE Trans. Comput.* **C-32** (1983) 75–89.

[9] C.P. Kruskal and M. Snir, Some results on multistage interconnection networks for multiprocesors, *Proc. 1982 Conf. on Information Sciences and Systems*, Princeton University, 1982.

[10] C.P. Kruskal and M. Snir, Optimal interconnection networks for parallel processors: The importance of being square, in: Y. Yemini, ed., *Current Advances in Distributed Computing and Communications* (Computer Science Press, Rockville, MD, 1987); see also *Proc. 11th Ann. Internat. Symp. on Computer Architecture*, Ann Arbor, MI, U.S.A. (1984) 91–98.

[11] D.H. Lawrie, Access and alignment of data in an array processor, *IEEE Trans. Comput.* **C-24** (1975) 1145–1155.

[12] D.S. Parker, Jr., Notes on shuffle/exchange-type switching networks, *IEEE Trans. Comput.* **C-29** (1980) 213–222.

[13] J.A. Patel, Performance of processor-memory interconnections for multiprocessors, *IEEE Trans. Comput.* **C-30** (1981) 771–780.

[14] M.C. Pease, The indirect binary n-cube microprocessor array, *IEEE Trans. Comput.* **C-26** (1977) 458–473.

[15] H.J. Siegel, Interconnection networks for SIMD machines, *Computer* **12** (1979) 57–65.

[16] H.J. Siegel and S.D. Smith, Study of multistage SIMD interconnection networks, *Proc. 5th Ann. Symp. on Computer Architecture*, California, U.S.A. (1978) 223–229.

[17] J.J. Siegel, *Interconnection Networks for Large-Scale Parallel Processing: Theory and Case Studies* (Lexington Books, Lexington, MA, 1985).

[18] B.W. Weide, A flexible packet-switching chip and its applications, *Proc. 19th Allerton Conf. on Communication, Control, and Computing*, Urbana, IL, U.S.A. 1981.

[19] C.L. Wu and T-Y. Feng, On a class of multistage interconnection networks, *IEEE Trans. Comput.* **C-29** (1980) 694–702.